AI Lifecycle Breakdown on GPU Platform (Granular Version)

|  |  |  |  |
| --- | --- | --- | --- |
| Stage | Technical Details | Platform-Specific Enhancements | Granular Tasks/Details |
| 1. Data Ingestion | Ingest structured/unstructured data from S3, APIs, Kafka, DBs. | NetApp S3 provides scalable, high-speed parallel I/O. | - Configure S3 buckets - Create ETL pipelines - Implement Kafka consumers |
| 2. Data Preparation | ETL pipelines, cleansing, deduplication, annotation. | Run:AI jobs for parallelized data prep; containers on OCP. | - Null handling - Data type normalization - Label encoding - Batch processing scripts |
| 3. Feature Engineering | Vectorization, embeddings, normalization, dimensionality reduction. | GPU acceleration for embeddings & AutoML feature selection. | - Use BERT embeddings - PCA dimensionality reduction - Feature scaling - AutoML ranking |
| 4. Model Selection | Choose transformer models, CNNs, RNNs, or LLMs. | Supports Hugging Face, PyTorch, TensorFlow on H200 GPUs. | - Load pre-trained models - Evaluate model configs - Select architectures |
| 5. Model Training | Distributed training with DDP, Horovod, DeepSpeed. | Run:AI orchestrates GPU sharing/scheduling across 24x H200 GPUs. | - Configure NCCL backend - Set mixed precision - Enable gradient checkpointing |
| 6. Model Evaluation | Validation metrics (accuracy, F1) on test sets or shadow mode. | On-demand jobs via OpenShift; tracked via Run:AI dashboards. | - Run confusion matrix - Evaluate precision/recall - Plot ROC-AUC |
| 7. Hyperparameter Tuning | Grid search, Bayesian opt, early stopping, sweeps. | Run:AI enables multiple parallel tuning jobs. | - Define hyperparam ranges - Launch Optuna/Ray Tune jobs - Log results in MLflow |
| 8. Model Packaging | Convert to ONNX, TorchScript; containerize via Docker. | Deploy containers on OpenShift with GPU access. | - Export to ONNX - Create Dockerfile - Build image - Push to registry |
| 9. Model Deployment | Batch, real-time, or edge deployment via REST, gRPC, or Kafka. | Integrated OpenShift CI/CD; GPU inference via Triton. | - Deploy Triton server - Expose service - Set auto-scaling rules |
| 10. Monitoring & Feedback | Drift detection, logging, performance dashboards. | NetApp logging + Run:AI workload reports + OCP observability. | - Set Prometheus alerts - Visualize GPU metrics - Track data drift stats |
| 11. Model Retraining | Trigger retraining on feedback, drift, or performance decay. | Run:AI automates job queueing + scheduled retraining. | - Define retrain trigger - Use existing pipeline - Version new model in registry |